Research Article Vol. 27, No. 6 | 18 Mar 2019 | OPTICS EXPRESS 8612

Optics EXPRESS

# Fourier ptychographic microscopy reconstruction with multiscale deep residual network

JIZHOU ZHANG,[1] TINGFA XU,[1,*] ZIYI SHEN,[1] YIFAN QIAO,[2] AND YIZHOU ZHANG[1]

[1] *School of Optics and Photonics, Beijing Institute of Technology, Beijing 100081, China*
[2] *Beijing 101 Middle School, Beijing 100081, China*
[*] *ciom_xtf1@bit.edu.cn*

**Abstract:** Fourier ptychographic microscopy (FPM) is a newly developed microscopic technique for large field of view, high-resolution and quantitative phase imaging by combining the techniques from ptychographic imaging, aperture synthesizing and phase retrieval. In FPM, an LED array is utilized to illuminate the specimen from different angles and the corresponding intensity images are synthesized to reconstruct a high-resolution complex field. As a flexible and low-cost approach to achieve high-resolution, wide-field and quantitative phase imaging, FPM is of enormous potential in biomedical applications such as hematology and pathology. Conventionally, the FPM reconstruction problem is solved by using a phase retrieval method, termed Alternate Projection. By iteratively updating the Fourier spectrum with low-resolution-intensity images, the result converges to a high-resolution complex field. Here we propose a new FPM reconstruction framework with deep learning methods and design a multiscale, deep residual neural network for FPM reconstruction. We employ the widely used open-source deep learning library PyTorch to train and test our model and carefully choose the hyperparameters of our model. To train and analyze our model, we build a large-scale simulation dataset with an FPM imaging model and an actual dataset captured with an FPM system. The simulation dataset and actual dataset are separated as training datasets and test datasets, respectively. Our model is trained with the simulation training dataset and fine tuned with the fine-tune dataset, which contains actual training data. Both our model and the conventional method are tested on the simulation test dataset and the actual test dataset to evaluate the performances. We also show the reconstruction result of another neural network-based method for comparison. The experiments demonstrate that our model achieves better reconstruction results and consumes much less time than conventional methods. The results also point out that our model is more robust under system aberrations such as noise and blurring (fewer intensity images) compared with conventional methods.

© 2019 Optical Society of America under the terms of the OSA Open Access Publishing Agreement

## 1. Introduction

Fourier ptychographic microscopy (FPM) is a newly developed microscopic technique for large field of view (FOV), high resolution and quantitative phase imaging [1–4]. FPM is developed from the lens-less imaging technique termed ptychography [5–7] and combines the concepts of aperture synthesizing [8–10] and phase retrieval [11–13]. The Fourier ptychography technique enlarges the equivalent numerical aperture (NA) of the system and achieves phase imaging without lasers by simply employing a LED array. The ptychography method illuminates the specimen with a focused beam in real space and records the diffraction patterns while scanning to different positions. Different from ptychography, FPM illuminates the specimen with angle-varied plane waves provided by a LED array. When illuminated with oblique plane waves, high-frequency information of the specimen is shifted into the passband of the objective lens. Similar to aperture synthesizing, FPM collects images containing different high-frequency information and stitches

#348562 https://doi.org/10.1364/OE.27.008612
Journal © 2019 Received 18 Oct 2018; revised 6 Dec 2018; accepted 15 Dec 2018; published 11 Mar 2019

them together in Fourier space to enlarge the passband of the optical system. A larger passband in Fourier space results in a larger equivalent NA and a higher resolution. Therefore, building a FPM with a low-NA microscope system achieves large FOV and high resolution at the same time [1, 4, 14]. Different from aperture synthesizing, FPM does not measure the phase directly at each illumination angle but applies the phase retrieval technique to recover the lost phase information.

FPM offers a flexible and low-cost approach to achieve high resolution, wide field and quantitative phase imaging without any mechanical moving components or phase measurements. Since the appearance of FPM, it has been used in many biomedical applications such as hematology [15] and pathology [16, 17]. Various modifications have also been implemented on both the system setup and the reconstruction method to improve FPM. Similar to ptychography, many system aberrations can be corrected in FPM, such as the aberration of objective lens [18, 19], the positions of LED elements [20, 21] and the defocus distance [19, 22]. To accelerate the image capture procedure, several strategies such as content awareness [23] and multiplexing [3,24,25] are introduced into FPM. Optimization theories like Gauss-Newton method [26], Wirtinger flow [27] and convex relaxation [28] are developed to improve the reconstruction speed and robustness. Innovative system designs for FPM such as lens-less system [29], reflective system [30], fluorescent system [31] and macroscopic system [32, 33] are also reported. These modifications show the great prospect of FPM in biomedical observation and clinical diagnosis. However, there are still some defects in the conventional FPM reconstruction methods such as low temporal resolution and poor robustness under system aberrations.

In this paper, we focus on solving the FPM reconstruction problem with deep learning. Deep learning is a machine learning technique that finds the correct mathematical mapping between input and output with deep neural networks (DNN) [34, 35]. Given large amounts of training data, deep neural networks can be trained to solve high-level and low-level computer vision problems. DNN based methods have been applied in image reconstruction [36–40] and phase retrieval [41, 42]. These works have achieved favourable results and expanded the application of DNN. Recently, methods that utilizing DNN to perform FPM reconstruction have be proposed [43–45] which perform well in reconstructing high-resolution images. However, these methods cannot recover phase images or focus on one type of sample. Inspired by these researches [36–45], we build a multiscale deep residual network termed Fourier ptychographic neural network (FPNN) to solve the FPM reconstruction problem. FPNN is able to reconstruct the high-resolution complex fields of many types of samples. To build a network with a reasonable size, we first synthetic the captured FPM intensity images into a complex filed (without iteration) and use the complex field as network input. The output of the network is also a complex field. We capture 400 animal tissue images and build a simulation dataset by performing FPM imaging model with the tissue images. We also build an actual dataset by capturing intensity images using a FPM system. The simulation dataset and actual dataset are separated as training datasets and test datasets respectively. Our model is trained with the simulation training dataset and finetuned with a finetune dataset which contains actual training data. Experiments show that the FPNN model performs a great reconstruction result on the simulation test dataset and the actual test dataset.

Comparing with traditional iteration-based reconstruction methods, FPNN achieves better results while consuming much less time. Once well trained, FPNN can perform high-quality FPM reconstruction rapidly by using the graphics processing unit (GPU). Experiments also show that FPNN can still reach a good reconstruction result when fewer intensity images are provided or system aberration exists, which is due to the excellent generalization ability of DNN. Shaowei Jiang et al. recently modeled FPM with neural network and solved the FPM reconstruction problem with back propagation [46]. They creatively solved the FPM reconstruction problem in the training process by forcing the network loss as zero. However, their method is still an iteration-based algorithm although the calculation is done with neural networks. Different from their solution, our model is a DNN that studies the non-linear relations between input and output by training with a large scale dataset. Our work is also inspired by Yair Rivenson et al. who propose a research [41] that solves phase retrieval with DNN. They report a new framework that reconstructs phase and amplitude images of the objects using only one hologram with an appropriately trained DNN. But the ground truth images they use to train the DNN are calculated by using the traditional iteration-based method. We employ another strategy to build the training dataset that using the captured high-resolution images as ground truth and generating simulated FPM intensity images with FPM imaging model. With our strategy, FPNN can achieve a better performance than traditional iteration-based methods and it is possible for us to quantitatively analyze the reconstruction results.

This paper is structured as follows. In Section 2, we discuss the imaging model and reconstruction model of FPM. In Section 3, we propose a new framework for FPM reconstruction by using DNN and describe the architecture of our network termed FPNN. In Section 4, we show the performance of FPNN and traditional FPM reconstruction methods. Finally, we conclude this paper with some summaries and discussions in Section 5.

## 2. Principle of Fourier ptychographic microscopy

The theories of FPM are mainly composed of two parts, the imaging model and the reconstruction model. The imaging model represents the process that plane waves pass through the microscope and image on the sensor. The reconstruction model represents the process that recovering the high-resolution complex field with the low-resolution intensity images. In practice, the imaging model is mainly used for simulation and only the reconstruction model is needed for recovering the complex field.

In FPM, a thin specimen can be represented by its transmission function $o(\mathbf{r})$, where $\mathbf{r} = (x, y)$ represents the 2D spatial coordinates. Assuming the LEDs are sufficiently far from the sample stage, the illumination waves are approximately oblique plane waves. For the $l$th LED, the wave vector can be formulated as

$$\mathbf{k}_l = \left(\frac{\sin\theta_{xl}}{\lambda}, \frac{\sin\theta_{yl}}{\lambda}\right) \quad (l = 1, 2, \cdots, N_{LED}), \tag{1}$$

where $(\theta_{xl}, \theta_{yl})$ denote the illumination angle for the $l$th LED and $\lambda$ is the wavelength. Illuminating the specimen using an oblique plane waves with a wave vector $\mathbf{k}_l$ is equivalent to shifting the specimen spectrum $O(\mathbf{k})$ to be centered around $\mathbf{k}_l$, expressed as

$$\mathcal{F}\{o(\mathbf{r})\exp(i\mathbf{k}_l\mathbf{r})\} = O(\mathbf{k} - \mathbf{k}_l) \tag{2}$$

When passing through the objective lens, the field is low-pass filtered by the objective lens with pupil function $P(\mathbf{k})$. The imaging model of FPM can be expressed as

$$I_{lc}(\mathbf{r}) = |g_{lc}(\mathbf{r})|^2 = \left|\mathcal{F}^{-1}\left\{P(\mathbf{k})O(\mathbf{k} - \mathbf{k}_l)\right\}\right|^2, \tag{3}$$

where $I_{lc}(\mathbf{r})$ denotes the intensity on sensor, $g_{lc}(\mathbf{r})$ denotes the complex field on sensor, $O(\mathbf{k} - \mathbf{k}_l)$ denotes the spectrum of the specimen illuminated by a plane wave with a wave vector $\mathbf{k}_l$, $\mathbf{k} = (k_x, k_y)$ represents the 2D frequency coordinates and $\mathcal{F}^{-1}$ denotes the inverse Fourier transform.

The principle of FPM reconstruction is to synthesize $g_{lc}(\mathbf{r})$ in Fourier space and get an estimation of high-resolution complex field $o_e(\mathbf{r}) = \mathcal{F}^{-1}\{O_e(\mathbf{k})\}$. However, the phase information is lost during the imaging process and only $I_{lc}(\mathbf{r})$ is recorded. To solve this problem, the phase retrieval method is introduced. The most classic reconstruction method, termed Alternate Projection (AP), is to iteratively estimate the complex field and update them with captured intensity images. One iteration can be formulated as

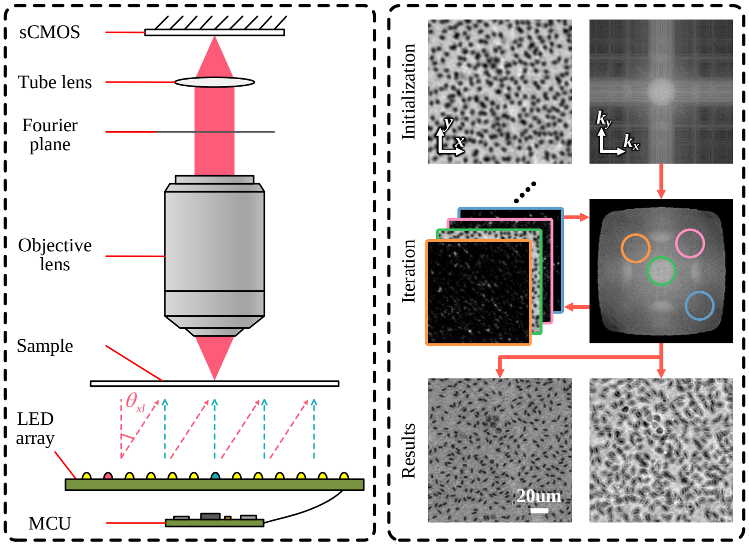
$$g_{le}(\mathbf{r}) = F^{-1}\left\{P(\mathbf{k})O_e(\mathbf{k}-\mathbf{k}_l)\right\}, \tag{4}$$

and

$$P(\mathbf{k})O_e(\mathbf{k}-\mathbf{k}_l) = F\left\{\frac{\sqrt{I_{lc}(\mathbf{r})}}{|g_{le}(\mathbf{r})|}g_{le}(\mathbf{r})\right\}. \tag{5}$$

Equation (4) is to estimate the complex fields corresponding to each illumination and Eq. (5) is to update the complex fields using the captured low-resolution intensity images. The operations in Eqs. (4) and (5) are repeated until the estimated spectrum converges and the iteration starts from an intensity-only guess of $g_{le}(\mathbf{r})$.

At last, by inverse Fourier transforming the estimated spectrum $O_e(\mathbf{k})$ to $o_e(\mathbf{r})$, the high-resolution intensity and phase are extracted from $o_e(\mathbf{r})$.



(a) (b)

Fig. 1. Schematic diagram of FPM. (a) The system setup of a FPM. (b) The reconstruction process of FPM.

Figure 1(a) shows the setup of a typical FPM system for capturing low-resolution intensity images. Figure 1(b) shows the process of the FPM reconstruction process of traditional iteration-based methods. The goal of this paper is to reconstruct high-resolution complex fields with a neural network instead of the FPM reconstruction model.

## 3. FPM reconstruction framework with FPNN

In this section, we propose a new framework to solve the FPM reconstruction problem. Our framework includes preparing the model input, building the dataset and, most important of all, the proposed FPNN model. We describe a method to synthetic low-resolution intensity images together as a complex field input for FPNN. We also describe how to build datasets to train FPNN with FPM imaging model. We are inspired by many image reconstruction neural networks when designing our model.

### 3.1. Model input

In a typical FPM setup, a LED array with hundreds of LEDs is used to provide illumination. In other words, the input data is a 3D image tensor with hundreds of channels. Generally, the channels in the first layer of a DNN should be dozens of times that of the input data to extract image features well. If the captured intensity images are directly used as input of the neural network, the network channels of each layer will be of hundreds or even thousands, which is out of reality. Not only that, the model will not be able to work on other systems with a different LED number if the images are directly inputted. To avoid these problems, the images are synthesized in Fourier space to form a complex field, formulated as

$$P(\mathbf{k})O_i(\mathbf{k}-\mathbf{k}_l) = F\left\{\sqrt{I_{lc}(\mathbf{r})}\right\}. \tag{6}$$

The operation in Eq. (6) is very similar to the update process of one iteration in Eq. (5). Therefore, Eq. (6) can be regarded as half of an iteration in traditional FPM reconstruction. The synthesized spectrum is inverse Fourier transformed to a complex field, expressed as

$$o_i(\mathbf{r}) = \mathcal{F}^{-1}\left\{O_i(\mathbf{k})\right\}. \tag{7}$$

Although $o_i(\mathbf{r})$ is a high-resolution complex field, there are many artifacts in it. Traditionally, the artifacts are eliminated gradually with dozens of times of iterations. In this paper, we use neural networks to finish the reconstruction next. Since the complex fields cannot be processed by most deep-learning libraries directly, the intensity and phase of $o_i(\mathbf{r})$ are extracted and combined as the input of the model.

As shown in Fig. 2, the synthetic input reduces the data channels of the neural network input and subsequent layers. Therefore, we are able to train the neural network on a GPU. Besides, with the synthetic input, if a trained network is applied to a different system setup, only the synthesis process needs to be revised.

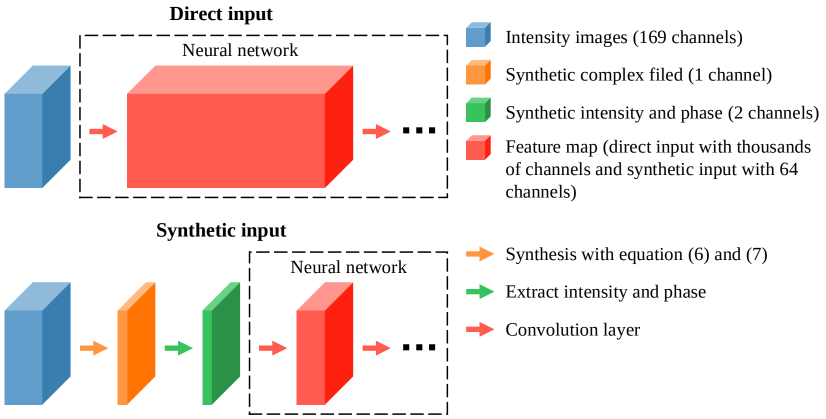


Fig. 2. Differences between direct input and synthetic input.

### 3.2. Building the datasets

To use a deep neural network, a large training dataset is required to train the network and a small test dataset is required to test the network performance. At the very beginning, we applied a strategy like the one proposed by Yair Rivenson et al. [41]. We captured dozens of groups of FPM low-resolution intensity images and used the FPM reconstruction results as the ground truth. The dataset was separated into a large part as the training dataset and a small part as the test dataset. The network converged fast with the training dataset and performs well on the test dataset. However, we find the problems using this strategy soon. Firstly, the intensity and phase of a specimen are highly related to each other in reality. With the actual data building the training dataset, the network learns the correlation between intensity and phase and the mapping of the FPM reconstruction is weakened. Secondly, the correlations between intensity and phase are not the same in different specimens. Training with actual data, the generalization ability of the network is weakened. At last, with the FPM reconstruction results as the ground truth, the network will never reach a better result than conventional FPM reconstruction methods.

To avoid the above problems, we generate datasets with simulations, like the strategies used in many deep learning image reconstruction projects. The original images serve as the ground truth and the degraded images serve as the model input. We capture 400 high-resolution images of different specimens as the intensity and phase in the FPM imaging model. We use these images randomly as intensity and phase to get 1600 high-resolution complex fields which are used as the ground truth. The low-resolution intensity input of the model is generated with the FPM imaging model using these high-resolution complex fields. Gaussian distribution noises with zero mean and standard deviation ranging from 0 to $3 \times 10^{-4}$ are added on the dataset images. The dataset is randomly cropped to generate a dataset with 25600 patches and then separated randomly as the training dataset with 23040 patches and the test dataset with 2560 patches. An example of building the training dataset is shown in Fig. 3.

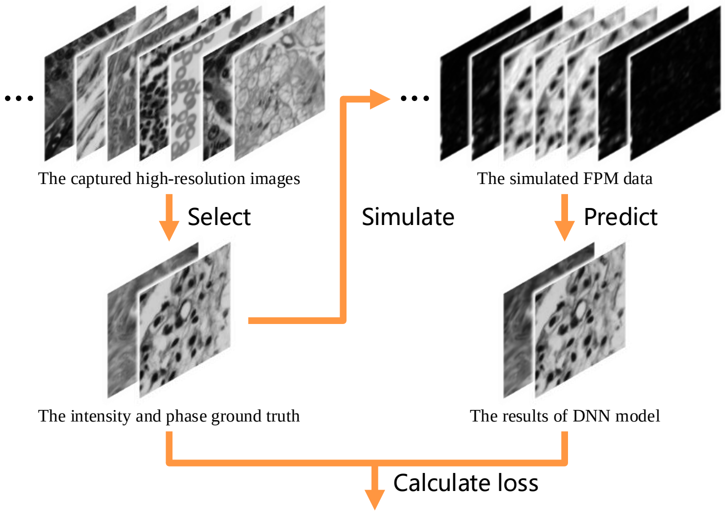


Fig. 3. An example of how to build and use the training dataset.

Besides the aforementioned simulation datasets, we build a finetune dataset. To prevent overfitting, the finetune dataset contains half simulation data and half actual data. We captured 50 sets of FPM low-resolution data and get the high-resolution results as the "ground truth" using the AP method. These data is randomly cropped to generate a dataset with 450 patches and then separated randomly as the training dataset with 400 patches and the test dataset with 50 patches. Then we randomly extract 400 patches of data from the simulation dataset to make up the finetune dataset together with the real data, because the real dataset is much smaller than the simulation dataset. With the finetune dataset to finetune the model, the model can learn the system aberrations which are unable to quantify and the performance of the model will be further improved. For lack of ground truth, the actual test dataset is only used for qualitative analysis.

As shown in Fig. 1(a), the system we use to capture images equips a 4× objective lens with a numerical aperture (NA) of 0.13. A scientific CMOS (sCMOS) camera with 2560×2160 pixels (6.5 um pixel size) is used to record the intensity images. The LED array (505 nm central wavelength) with 169 elements is placed 100 mm below the specimen for providing angle-varied illumination. The illumination NA of the LED array is 0.4455 and a micro control unit (MCU) is used to control the LED array.

### 3.3. Reconstruction with FPNN

As mentioned before, Shaowei Jiang et al. report a method that models FPM with neural networks and solves the FPM reconstruction problem with back propagation [46]. They creatively covert the FPM reconstruction problem to training a neural network. However, they only use the neural network to calculate the gradient and minimize the loss like other optimization methods [47]. Their method is still an iteration-based algorithm. With their method, every time to perform the reconstruction, the training process is carried out. Different from their strategy, we build our model using DNN and train the model to study the non-linear relations between input and output with a large-scale dataset. Once the DNN is trained, the reconstruction process can be rapidly performed by predicting with the DNN.

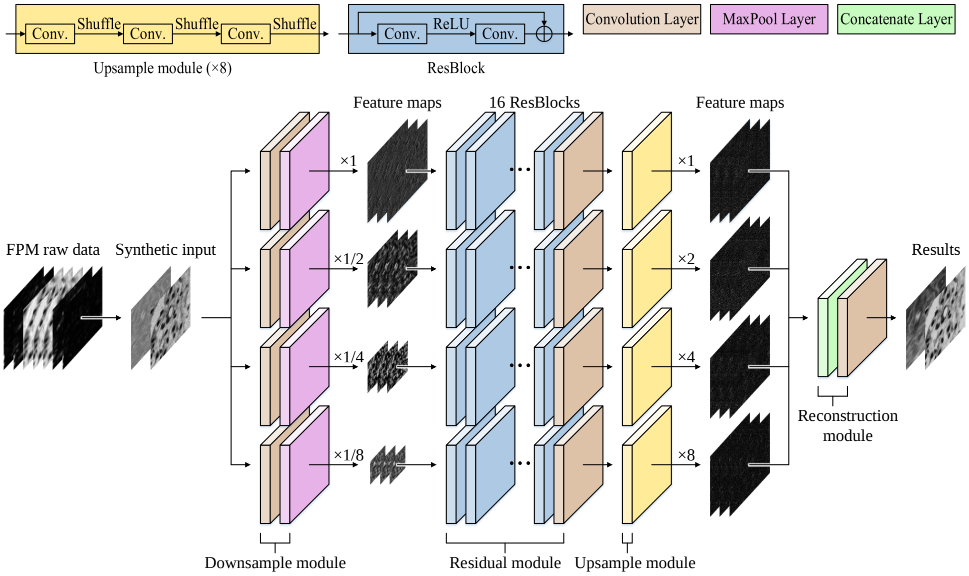


Fig. 4. Model architecture of FPNN. The architecture of a residual block and the architecture of an upsample module are shown on the right side.

Considering a set of FPM low-resolution intensity images as $\mathbf{Z}$, we get the 2-channels input tensor $\mathbf{Y}$ by synthesizing $\mathbf{Z}$, $\mathbf{Y} = Syn(\mathbf{Z})$, which is defined in Eqs. (6) and (7). The goal of our model is to reconstruct from $\mathbf{Y}$ a 2-channels tensor $F(\mathbf{Y})$ that is as similar as possible to the ground truth $\mathbf{X}$. To achieve the goal, we design a multiscale deep residual neural network termed FPNN. Although designed for FPM reconstruction, FPNN has the potential for other image reconstruction applications such as image super resolution and imaging deblurring. The architecture of FPNN is detailed in Fig. 4 with different color blocks representing different network layers. From the view of function, FPNN consists of four parts.

1) Downsample module. The synthesized 2-channels input is simultaneously inputted to four

convolutional layers, creating four data flows. The four convolutional layers take 2-channels images as input and output 64-channels feature maps. The output feature maps are then downsampled ×1, ×2, ×4, ×8 by four max pooling (MaxPool) layers. This multiscale architecture enables the network to suppress the interference artifacts and reconstruct the high-resolution complex field with different feature sizes.

2) Residual module. The output of a MaxPool layer is followed by 16 residual blocks and a convolutional layer. Each residual block is composed of two convolutional layers and one rectified linear unit (ReLU) activation layer sandwiched in the middle. Residual blocks create information flows between the input and output of the blocks, which greatly enhances the performance and convergence speed of the neural network. We use the residual block proposed by Bee Lim et al. [40] which is more compact and performs better comparing with residual blocks in other models such as ResNet [48] and SRResNet [39]. Sixteen basic residual blocks and one convolutional layer make up a large residual block. The large residual block takes a 64-channels feature map as input and outputs a 64-channels feature map.

3) Upsample module. After the residual module is the upsample module which upsamples the feature map to the same size with the synthetic input. The upsample module architecture is related to the downsample times. For example, the ×8 upsample module is composed of three convolutional layers and three pixel shuffle (PixelShuffle) layers sandwiched together. The convolutional layers in upsample module take 64-channels feature maps as input and output 256-channels feature maps. The PixelShuffle layers in upsample module take 256-channels feature maps as input and output 64-channels feature maps while upsampling the size ×2. The upsample module is ended with a convolutional layer that takes a 64-channels feature map as input and outputs a 64-channels feature map.

4) Reconstruction module. After the upsample module, four 64-channels feature maps are concatenated together to be a 256-channels feature map. This concatenate operation synthesizes all features from four scales together. At last, a convolutional layer takes in the 256-channels feature map and outputs a 2-channels feature map which is composed of the high-resolution intensity and phase.

## 4. Experiments

In this section, we show the results of our simulations and experiments. The FPNN is already pretrained with the simulation training dataset and finetuned with the finetune dataset before the test. We finetune FPNN with the purpose of improving its performance on actual data. The hyperparameters of FPNN are carefully adjusted to reach the best performance in the training process. Specifically, we set the learning rate as $1 \times 10^{-4}$, the kernel size as 3, the batch size as 16 and the patch size as 192. We pretrain FPNN with 300 epochs and finetune the pretrained FPNN with 100 epochs. We use $L_1$ loss function to constraint the training. We perform FPNN and AP on the simulation test dataset under a series of noise levels. The performance and speed of different reconstruction methods are also compared. We also show the performance of FPNN with fewer intensity images. At last, we show the reconstruction results of FPNN and AP on the actual dataset. To be clear, all results of FPNN in subsection 4.1 and 4.3 are test on a same well trained model and the models in subsection 4.2 are trained respectively because their training datasets are completely different from each other. All methods are implemented with Python and run on a NVIDIA GTX 1080Ti graphics card.

### *4.1. Performance under noise*

To evaluate the effectiveness of FPNN, we perform FPM reconstruction with FPNN and AP on the simulation test dataset and compare the results of these two methods. Under actual experimental conditions, the captured intensity images are contaminated by imaging noise which greatly affects the reconstruction results. To simulate the actual conditions, Gaussian distribution noises with

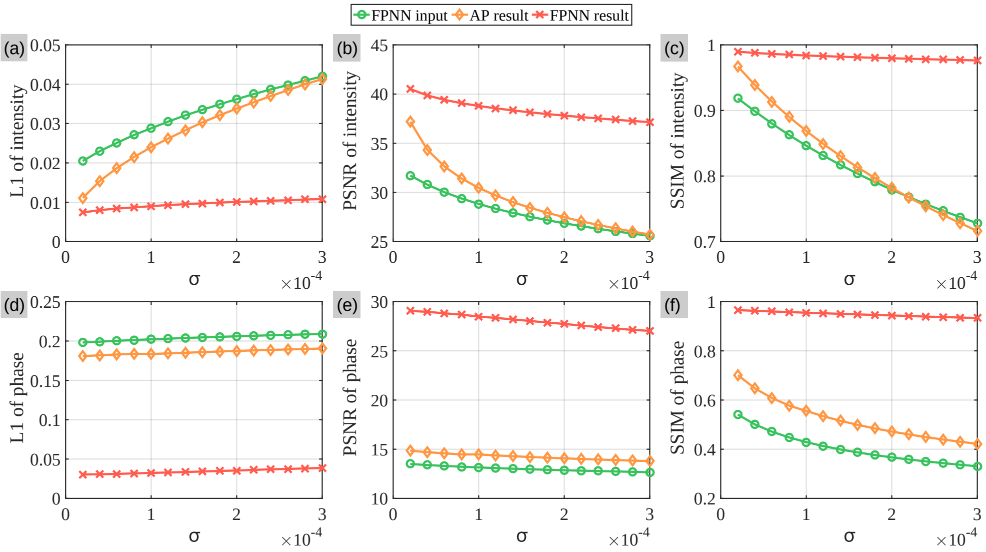


Fig. 5. Evaluation of reconstruction results under a series of noise levels. (a)-(c) show the L1 loss, PSNR and SSIM of reconstructed intensity images under noise. (d)-(f) show the L1 loss, PSNR and SSIM of reconstructed phase images under noise.

zero mean and standard deviation ranging from $2 \times 10^{-5}$ to $3 \times 10^{-4}$ are added on the test dataset respectively. The FPNN method is already trained on the training dataset before test and dataset images are synthesized as the complex field before input to the FPNN model. Fig. 5 shows the Manhattan distance (L1 loss), peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) of AP and FPNN under different noise levels. Besides, Fig. 6 shows some example reconstruction results of these two methods as well as Shaowei Jiang's method [46]. The central low-resolution intensity image, the input of FPNN and the ground truth are also shown. The L1 loss, PSNR and SSIM of these example images are also shown in the figure. Because Jiang's method costs too much time to finish one reconstruction, we are not able to test it on the whole test dataset. As the plots and example results show, FPNN's results have more details and fewer artifacts comparing with AP and Jiang's method. The phase reconstruction results of AP and Jiang's method suffers from the deviation of gray-scale value while the phase results of FPNN are much more accurate. It can be concluded that FPNN performs better and more robust than AP and Jiang's method under noise. Table 1 shows the time consumption of all methods with different patch sizes. Benefiting from the end-to-end structure and GPU computing, FPNN achieves a much quick reconstruction speed, which will promote the application of FPM in dynamic observation.

Table 1. Time consumption of reconstructing a patch

| Patch size | AP[a] | FPNN[b] | Jiang's method[c] |
|---|---|---|---|
| 128 | 1.91s | 34.6ms | 6.57s |
| 192 | 2.63s | 55.9ms | 4.97min |
| 256 | 4.05s | 121ms | 15.8min |
| 512 | 12.7s | 431ms | 4.06h |

[a] The AP method is implemented with Python.
[b] The FPNN method is implemented with Python and PyTorch.
[c] Jiang's method is modified from the open source code [46] which is implemented with Python and Tensorflow.

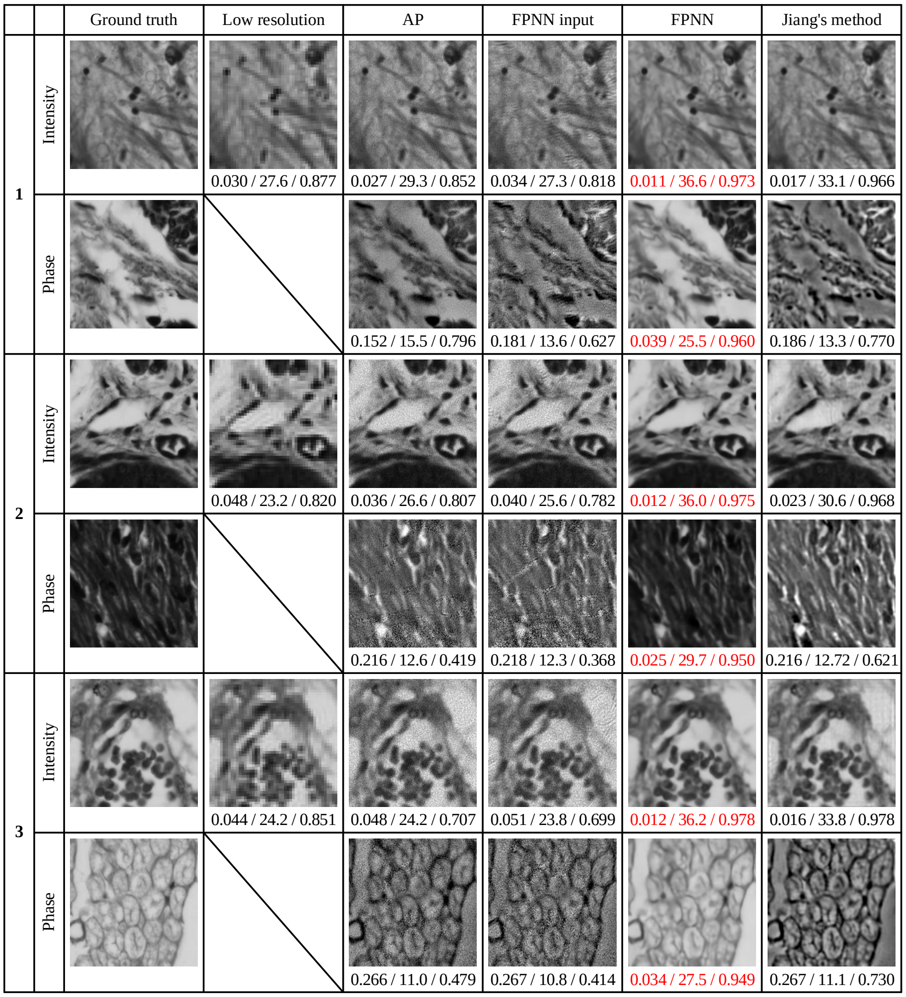


Fig. 6. Example reconstruction results of AP, FPNN and Shaowei Jiang's method on the simulation test dataset. Gaussian distribution noises with zero mean and standard deviation of $1 \times 10^{-4}$, $2 \times 10^{-4}$ and $3 \times 10^{-4}$ are added on group 1, 2 and 3.

### 4.2. Performance with fewer intensity images

As discussed in Section 1, more than one hundred low-resolution intensity images are captured and synthesized together in FPM reconstruction, which limits the temporal resolution of FPM. Reducing the number of intensity images, the final resolution of reconstruction results will decrease, which is equivalent to add blurring to the final results. If high-resolution intensity and phase can be reached with fewer low-resolution images, the temporal resolution of FPM will greatly increase. To test the performance of FPNN with fewer intensity images, we compare the reconstruction results of FPNN methods with 9, 25, 49, 81, 121, 169 images and the result of AP with 169 images. Gaussian distribution noises with zero mean and $1 \times 10^{-4}$ standard deviation is added on the test dataset. The FPNN model is only trained once with the simulation training dataset which is composed of 169-images data. The L1 loss, PSNR and SSIM of the reconstruction results are shown in Fig. 7. The experiment demonstrates that FPNN can still reach a usable reconstruction result when greatly reducing the captured intensity images. This

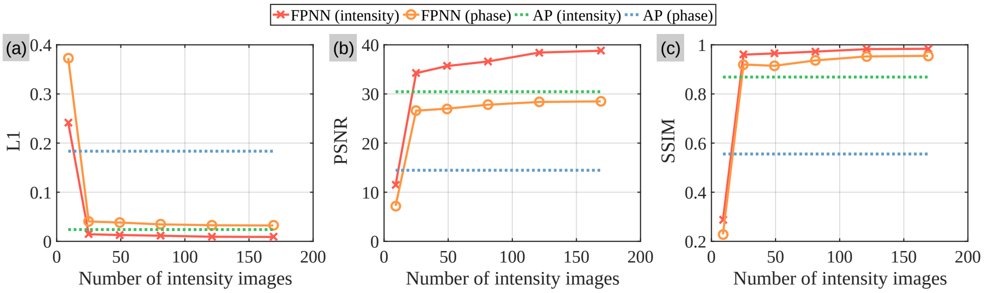


Fig. 7. Evaluation of reconstruction results with fewer intensity images. (a)-(c) show the L1 loss, PSNR and SSIM of reconstruction results respectively. The L1 loss, PSNR and SSIM of AP are always measured with 169 images.

characteristic of FPNN is very useful in conditions that requiring high temporal resolution. This experiment also proves that a pretrained FPNN model can be directly applied to a different FPM system and still achieves good reconstruction results.

### 4.3. *Reconstruction results on the actual dataset*

As mentioned in Subsection 3.2, we build a dataset using actually captured low-resolution images. We use the smaller part of this dataset to validate the performance of FPNN on the actual system. Because only low-resolution intensity images are captured, the ground truth of this dataset is nonexistent. Figure 8 shows some example reconstruction results of FPNN and AP on this dataset. As shown in the figure, FPNN achieves great reconstruction results comparing with AP and the results of FPNN are even cleaner than AP in some cases.

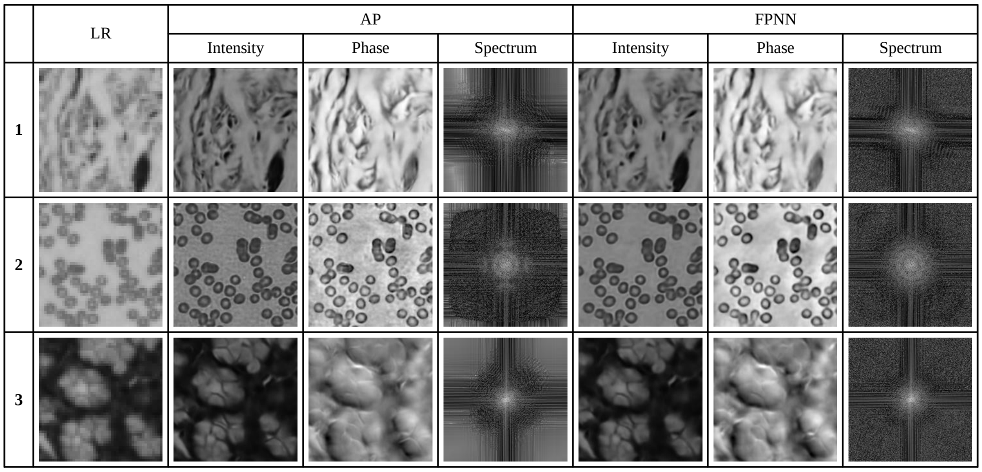


Fig. 8. Example reconstruction results of AP and FPNN on actual dataset.

## 5. Conclusion and discussion

In this paper, we propose a multiscale deep residual neural network termed FPNN for Fourier ptychographic microscopy reconstruction. In FPNN, the objects are treated as 2-channels feature maps with intensity and phase separated. To control the model size, the captured intensity images are synthesized as a 2-channels feature map before input to FPNN. We design four data flows with tandem residual blocks in FPNN to suppress the interference artifacts and reconstruct the high-resolution complex field with different feature sizes. All feature maps in four data flows are then concatenated together to reconstruct the final result. We employ the widely used open-source deep learning library PyTorch to train and test the model.

We carry out experiments to compare the performance of FPNN and AP under different noise levels. The results show that FPNN reaches better reconstruction results while consuming much less time. We also test the performance of FPNN with fewer intensity images. It can be concluded from the results that the reduction of intensity images has a much smaller impact on FPNN than AP. FPNN can still reach a usable reconstruction result while greatly reduce the required intensity images. With FPNN to recover the high-resolution intensity and phase, FPM can perform much better and faster. It is also reasonable to believe that our model can be used in other phase retrieval problems.

As shown in this paper, it is superior to use the deep neural network to perform FPM reconstruction. It would be a future work to find a better model architecture or a better training method. Here we propose some possible directions. First, a simple loss function can not perfectly constraint the training process and a loss with more statistical information may lead to better results. Second, although our FPNN performs better than the AP method under noise, it is trained using ideal simulation data without noise or other system aberrations. If the training dataset contains different kinds of system aberrations, the FPNN would be more robust.

## Funding

National Natural Science Foundation of China (NSFC) (61527802, 61371132, 61471043).

## Disclosures

The authors declare that there are no conflicts of interest related to this article.

## References

1. G. Zheng, R. Horstmeyer, and C. Yang, “Wide-field, high-resolution Fourier ptychographic microscopy,” Nat. Photonics **7**, 739–745 (2013).
2. X. Ou, R. Horstmeyer, C. Yang, and G. Zheng, “Quantitative phase imaging via Fourier ptychographic microscopy.” Opt. Lett. **38**, 4845–4848 (2013).
3. L. Tian, X. Li, K. Ramchandran, and L. Waller, “Multiplexed coded illumination for Fourier Ptychography with an LED array microscope,” Biomed. Opt. Express **5**, 2376 (2014).
4. K. Guo, S. Dong, and G. Zheng, “Fourier Ptychography for Brightfield , Phase ,Darkfield,Reflective,Multi-Slice, and Fluorescence Imaging,” IEEE journal selected topics Quantam Electron. **22**, 1–12 (2016).
5. H. M. Faulkner and J. M. Rodenburg, “Movable aperture lensless transmission microscopy: A novel phase retrieval algorithm,” Phys. Rev. Lett. **93**, 023903 (2004).
6. J. M. Rodenburg, A. C. Hurst, A. G. Cullis, B. R. Dobson, F. Pfeiffer, O. Bunk, C. David, K. Jefimovs, and I. Johnson, “Hard-X-ray lensless imaging of extended objects,” Phys. Rev. Lett. **98**, 17–21 (2007).
7. J. M. Rodenburg, “Ptychography and related diffractive imaging methods,” Adv. Imaging Electron Phys. **150**, 87–184 (2008).
8. V. Mico, Z. Zalevsky, P. García-Martínez, and J. García, “Synthetic aperture superresolution with multiple off-axis holograms,” J. Opt. Soc. Am. A **23**, 3162–3170 (2006).
9. T. R. Hillman, T. Gutzler, S. A. Alexandrov, and D. D. Sampson, “High-resolution, wide-field object reconstruction with synthetic aperture Fourier holographic optical microscopy,” Opt. Express **17**, 7873–7892 (2009).
10. T. Gutzler, T. R. Hillman, S. A. Alexandrov, and D. D. Sampson, “Coherent aperture-synthesis, wide-field, high-resolution holographic microscopy of biological tissue,” Opt. Lett. **35**, 1136–1138 (2010).
11. J. R. Fienup, “Phase retrieval algorithms: a comparison,” Appl. Opt. **21**, 2758 (1982).
12. A. M. Maiden and J. M. Rodenburg, “An improved ptychographical phase retrieval algorithm for diffractive imaging,” Ultramicroscopy **109**, 1256–1262 (2009).
13. I. Waldspurger, A. D’Aspremont, and S. Mallat, “Phase recovery, MaxCut and complex semidefinite programming,” Math. Program. **149**, 47–81 (2013).
14. X. Ou, R. Horstmeyer, G. Zheng, and C. Yang, “High numerical aperture Fourier ptychography: principle, implementation and characterization,” Opt. Express **23**, 3472 (2015).
15. J. Chung, X. Ou, R. P. Kulkarni, and C. Yang, “Counting white blood cells from a blood smear using fourier ptychographic microscopy,” PLoS ONE **10**, 1–10 (2015).

16. A. Williams, J. Chung, X. Ou, G. Zheng, S. Rawal, Z. Ao, R. Datar, C. Yang, and R. Cote, “Fourier ptychographic microscopy for filtration-based circulating tumor cell enumeration and analysis,” J. Biomed. Opt. **19**, 066007 (2014).
17. R. Horstmeyer, X. Ou, G. Zheng, P. Willems, and C. Yang, “Digital pathology with Fourier ptychography,” Comput. Med. Imaging Graph. **42**, 38–43 (2015).
18. X. Ou, G. Zheng, and C. Yang, “Embedded pupil function recovery for Fourier ptychographic microscopy,” Opt. Express **22**, 4960 (2014).
19. Z. Bian, S. Dong, and G. Zheng, “Adaptive system correction for robust Fourier ptychographic imaging,” Opt. Express **21**, 32400 (2013).
20. J. Sun, Q. Chen, Y. Zhang, and C. Zuo, “Efficient positional misalignment correction method for Fourier ptychographic microscopy,” Biomed. Opt. Express **7**, 1336 (2016).
21. J. Zhang, T. Xu, J. Liu, S. Chen, and X. Wang, “Precise Brightfield Localization Alignment for Fourier Ptychographic Microscopy,” IEEE Photonics J. **10**, 1–13 (2018).
22. L. Tian and L. Waller, “3D intensity and phase imaging from light field measurements in an LED array microscope,” Optica **2**, 104 (2015).
23. L. Bian, J. Suo, G. Situ, G. Zheng, F. Chen, and Q. Dai, “Content adaptive illumination for Fourier ptychography,” Opt. Lett. **39**, 6648 (2014).
24. S. Dong, R. Shiradkar, P. Nanda, and G. Zheng, “Spectral multiplexing and coherent-state decomposition in Fourier ptychographic imaging,” Biomed. Opt. Express **5**, 1757 (2014).
25. J. Zhang, T. Xu, S. Chen, and X. Wang, “Efficient Colorful Fourier Ptychographic Microscopy Reconstruction with Wavelet Fusion,” IEEE Access **6**, 31729–31739 (2018).
26. R. Horstmeyer, G. Zheng, X. Ou, and C. Yang, “Modeling extensions of fourier ptychographic microscopy,” Microsc. Microanal. **20**, 370–371 (2014).
27. L. Bian, J. Suo, G. Zheng, K. Guo, F. Chen, and Q. Dai, “Fourier ptychographic reconstruction using Wirtinger flow optimization,” Opt. Express **23**, 4856 (2015).
28. R. Horstmeyer, R. Y. Chen, X. Ou, B. Ames, J. A. Tropp, and C. Yang, “Solving ptychography with a convex relaxation,” New J. Phys. **17**, 53044 (2015).
29. W. Luo, A. Greenbaum, Y. Zhang, and A. Ozcan, “Synthetic aperture-based on-chip microscopy,” Light. Sci. & Appl. **4**, e261 (2015).
30. S. Pacheco, G. Zheng, and R. Liang, “Reflective Fourier ptychography,” J. Biomed. Opt. **21**, 026010 (2016).
31. S. Dong, P. Nanda, R. Shiradkar, K. Guo, and G. Zheng, “High-resolution fluorescence imaging via pattern-illuminated Fourier ptychography,” Opt. Express **22**, 20856 (2014).
32. S. Dong, R. Horstmeyer, R. Shiradkar, K. Guo, X. Ou, Z. Bian, H. Xin, and G. Zheng, “Aperture-scanning Fourier ptychography for 3D refocusing and super-resolution macroscopic imaging,” Opt. Express **22**, 13586 (2014).
33. J. Holloway, M. S. Asif, M. K. Sharma, N. Matsuda, R. Horstmeyer, O. Cossairt, and A. Veeraraghavan, “Toward Long-Distance Subdiffraction Imaging Using Coherent Camera Arrays,” IEEE Transactions on Comput. Imaging **2**, 251–265 (2016).
34. Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” Nature **521**, 436–444 (2015).
35. J. Schmidhuber, “Deep Learning in neural networks: An overview,” Neural Networks **61**, 85–117 (2015).
36. C. Dong, C. C. Loy, K. He, and X. Tang, “Learning a Deep Convolutional Network for Image Super-Resolution,” in Proceedings of the European conference on computer vision (ECCV), vol. 8689 (2014), pp. 184–199.
37. J. Kim, J. K. Lee, and K. M. Lee, “Accurate Image Super-Resolution Using Very Deep Convolutional Networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (IEEE, 2016), pp. 1646–1654.
38. Y. Tai, J. Yang, and X. Liu, “Image Super-Resolution via Deep Recursive Residual Network,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2017-Janua (IEEE, 2017), pp. 2790–2798.
39. C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (IEEE, 2017), pp. 105–114.
40. B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, “Enhanced Deep Residual Networks for Single Image Super-Resolution,” in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), vol. 2017-July (IEEE, 2017), pp. 1132–1140.
41. Y. Rivenson, Y. Zhang, H. Gunaydin, D. Teng, and A. Ozcan, “Phase recovery and holographic image reconstruction using deep learning in neural networks,” Light. Sci. & Appl. **1**, 1–30 (2017).
42. A. Sinha, J. Lee, S. Li, and G. Barbastathis, “Lensless computational imaging through deep learning,” Optica **4**, 1117 (2017).
43. A. Kappeler, S. Ghosh, J. Holloway, O. Cossairt, and A. Katsaggelos, “Ptychnet: CNN based fourier ptychography,” in IEEE International Conference on Image Processing (ICIP), (IEEE, 2017), pp. 1712–1716.
44. T. Nguyen, Y. Xue, Y. Li, L. Tian, and G. Nehmetallah, “Deep learning approach for Fourier ptychography microscopy,” Opt. Express **26**, 26470 (2018).
45. Y. F. Cheng, M. Strachan, Z. Weiss, M. Deb, D. Carone, and V. Ganapati, “Illumination pattern design with deep learning for single-shot fourier ptychographic microscopy,” arXiv preprint arXiv:1810.03481 (2018).
46. S. Jiang, K. Guo, J. Liao, and G. Zheng, “Solving Fourier ptychographic imaging problems via neural network

modeling and TensorFlow," Biomed. Opt. Express **9**, 3306 (2018).

47. L.-H. Yeh, J. Dong, J. Zhong, L. Tian, M. Chen, G. Tang, M. Soltanolkotabi, and L. Waller, "Experimental robustness of Fourier ptychography phase retrieval algorithms," Opt. Express **23**, 33214 (2015).
48. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (IEEE, 2016), pp. 770–778.

211.68.2.209 {ts '2019-03-14 02:16:53'}